(12) UK Patent Application (19) GB (11) 2 247 758 (13) A

(43) Date of A publication 11.03.1992

(21) Application No 9118335.0

(22) Date of filing 27.08.1991

(30) Priority data
(31) 02225665 (32) 28.08.1990 (33) JP

(71) Applicant
Kabushiki Kaisha Toshiba

(incorporated in Japan)

72 Horikawa-cho, Saiwai-ku, Kawasaki-shi, Kanagawa-ken, Japan

(72) Inventors
Kanee Kimura
Kazuhiro Kawagome

(74) Agent and/or Address for Service
Marks & Clerk
57-60 Lincoln's Inn Fields, London, WC2A 3LS, United Kingdom

(51) INT CL[5]
G06F 15/16 9/38

(52) UK CL (Edition K)
G4A AMP

(56) Documents cited
GB 2216306 A EP 0086601 A2

(58) Field of search
UK CL (Edition K) G4A AMP APP
INT CL[5] G06F 9/38 15/16
On-line databases : W. P. I.

(54) Controlling indivisible operation in parallel processing system

(57) In a parallel processing system having a plurality of processors 1 (2, Fig 1), when an instruction word read out from a shared memory 3 and having control bits *a*, *b* associated with an indivisible operation is received by an instruction decoder 11, the control bits are supplied to an indivisible-operation control circuit 7. If the control bits have values representing execution of the indivisible operation, the circuit 7 controls the indivisible operation via a squencer 12 in accordance with the set/reset state of a flip-flop 13, which represents whether execution of the indivisible operation is assured, and with the level state of an indivisible-operation control line 9 to which all the processors are coupled. The indivisible operation is used to inhibit execution of asynchronous processing caused by an interrupt while a program is running.

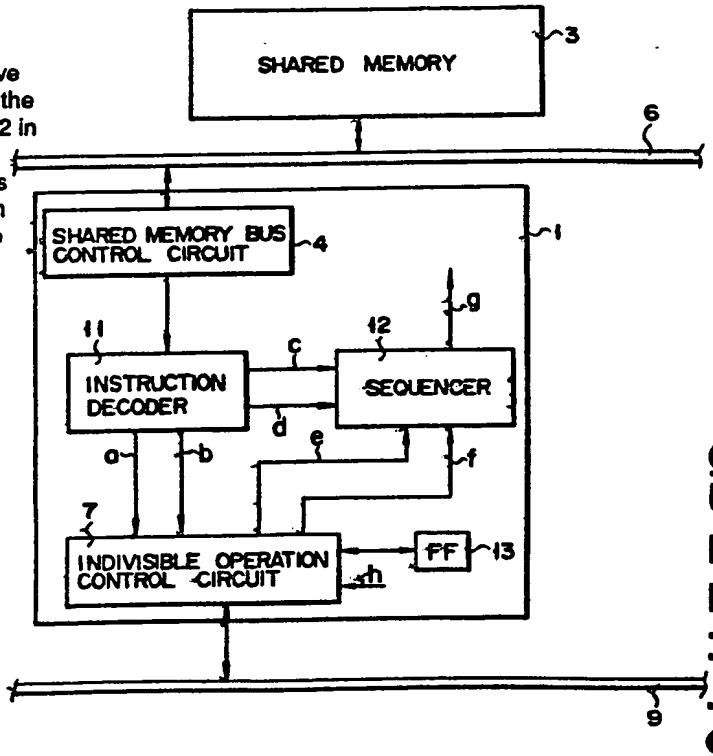

FIG. 3

GB 2 247 758 A

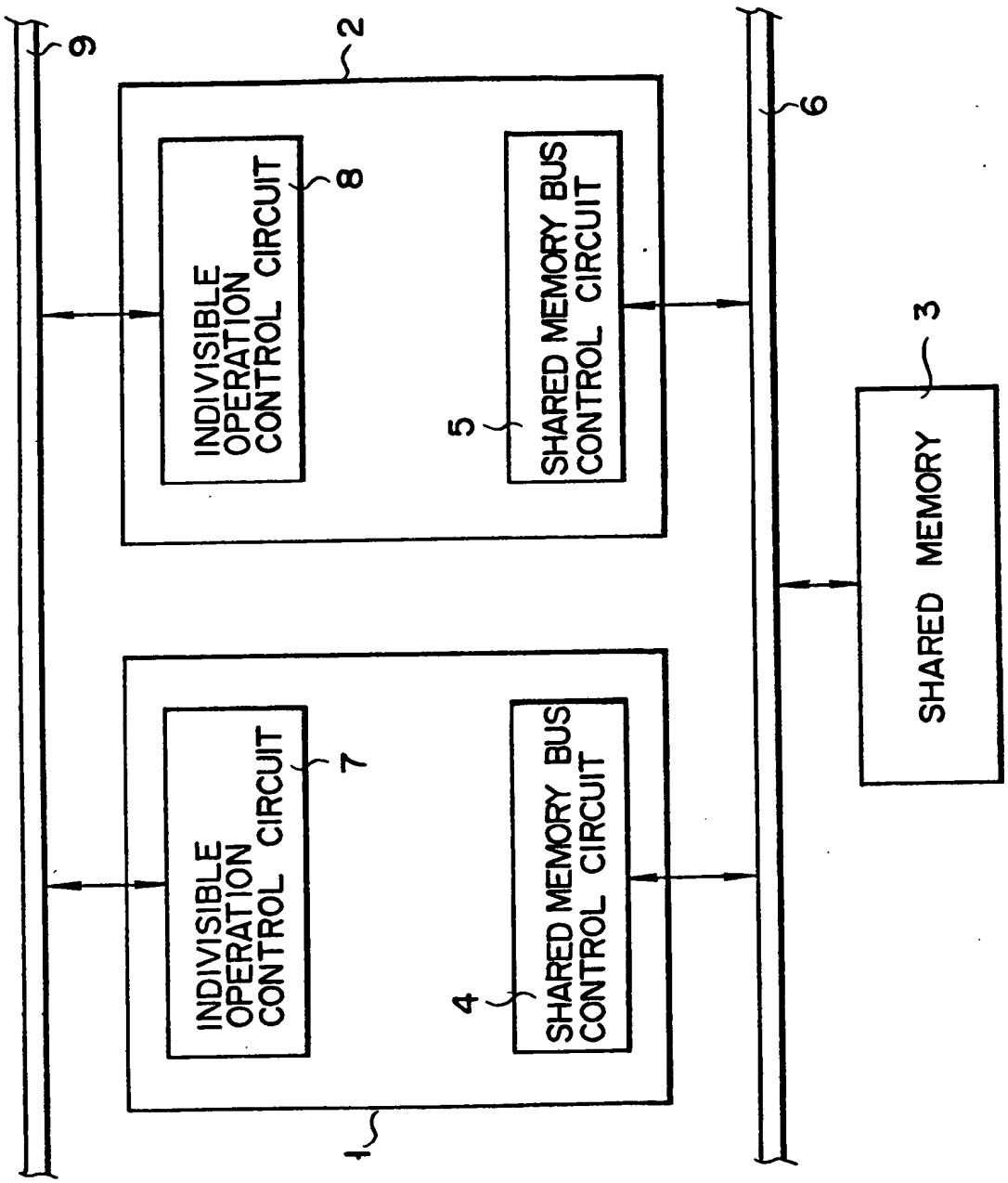
9
2
INDIVISIBLE OPERATION CONTROL CIRCUIT
8
5
SHARED MEMORY BUS CONTROL CIRCUIT
6
3
SHARED MEMORY
INDIVISIBLE OPERATION CONTROL CIRCUIT
7
4
SHARED MEMORY BUS CONTROL CIRCUIT
1

FIG. 1

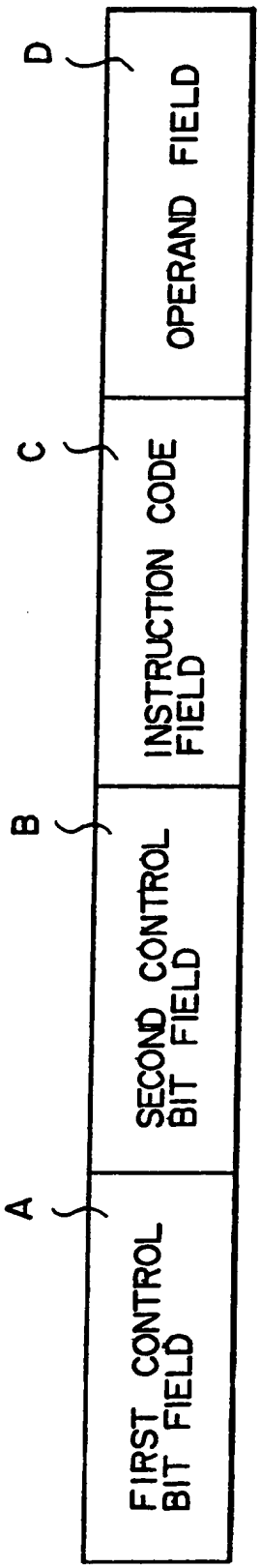
A
FIRST CONTROL BIT FIELD
B
SECOND CONTROL BIT FIELD
C
INSTRUCTION CODE FIELD
D
OPERAND FIELD

FIG. 2

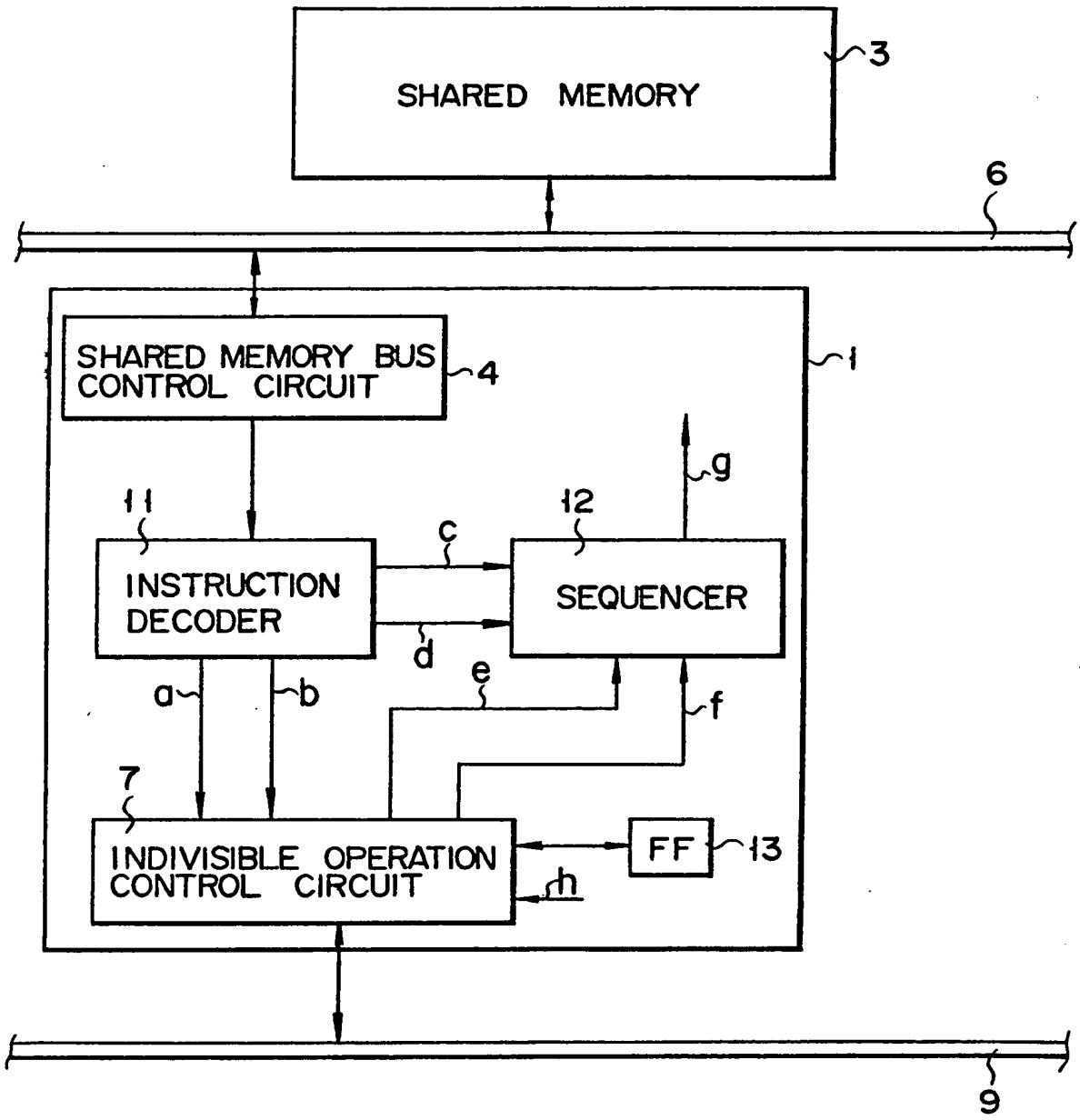
SHARED MEMORY
3
6
SHARED MEMORY BUS CONTROL CIRCUIT
4
1
11
12
g
c
INSTRUCTION DECODER
SEQUENCER
d
e
a
b
f
7
INDIVISIBLE OPERATION CONTROL CIRCUIT
FF
13
h
9

FIG. 3

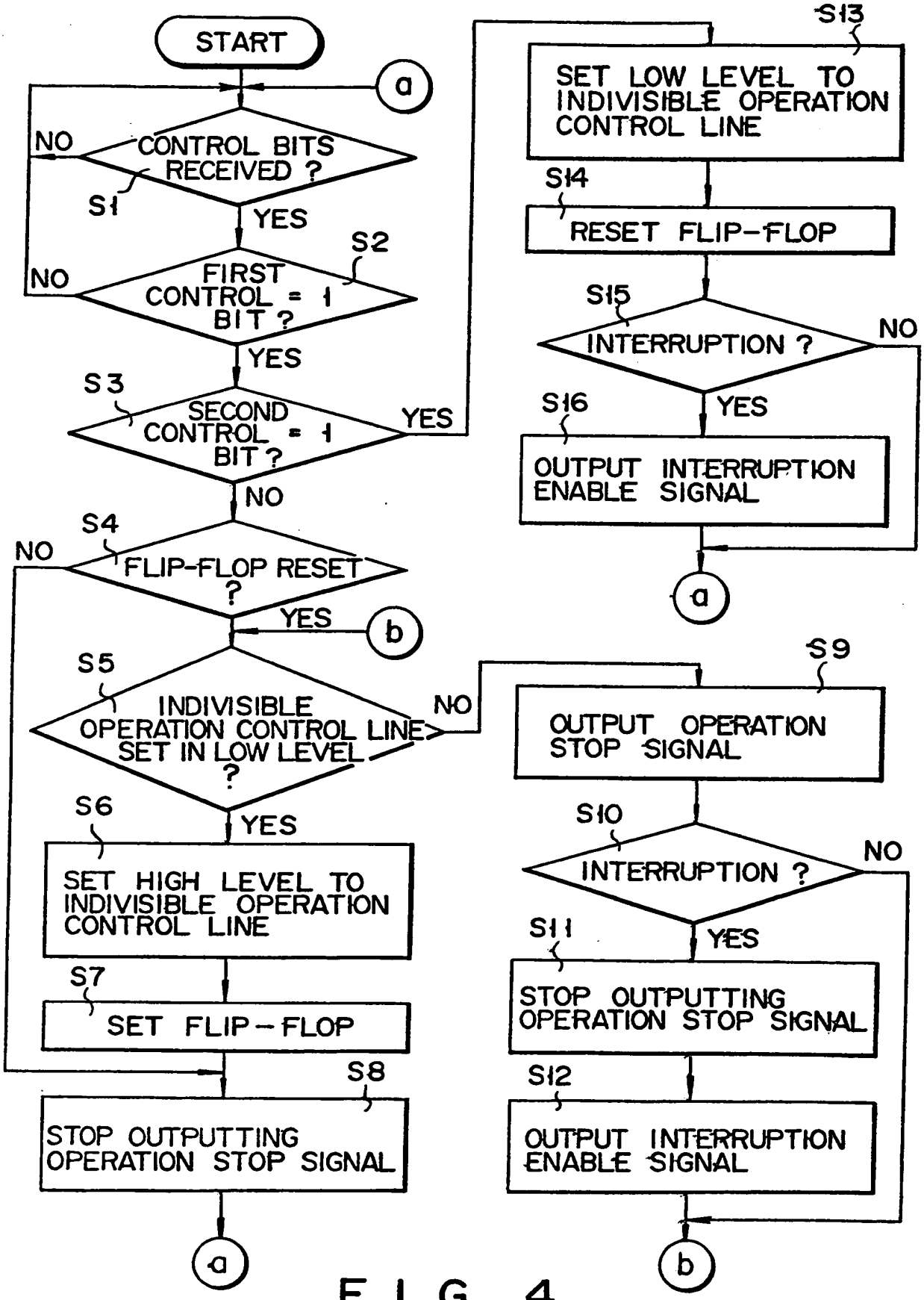
START
a
CONTROL BITS RECEIVED ?
NO
S1
YES
FIRST CONTROL = 1 BIT ?
S2
NO
YES
SECOND CONTROL = 1 BIT ?
S3
YES
NO
FLIP-FLOP RESET ?
S4
NO
YES
b
INDIVISIBLE OPERATION CONTROL LINE SET IN LOW LEVEL ?
S5
NO
YES
SET HIGH LEVEL TO INDIVISIBLE OPERATION CONTROL LINE
S6
SET FLIP-FLOP
S7
STOP OUTPUTTING OPERATION STOP SIGNAL
S8
a
SET LOW LEVEL TO INDIVISIBLE OPERATION CONTROL LINE
S13
RESET FLIP-FLOP
S14
INTERRUPTION ?
S15
NO
YES
OUTPUT INTERRUPTION ENABLE SIGNAL
S16
a
OUTPUT OPERATION STOP SIGNAL
S9
INTERRUPTION ?
S10
NO
YES
STOP OUTPUTTING OPERATION STOP SIGNAL
S11
OUTPUT INTERRUPTION ENABLE SIGNAL
S12
b

FIG. 4

"METHOD AND APPARATUS FOR CONTROLLING INDIVISIBLE OPERATION IN PARALLEL PROCESSING SYSTEM"

The present invention relates to a method and apparatus for controlling an indivisible operation in a parallel processing system.

In a parallel processing system, for example, a shared resource such as an input/output unit or a memory is often subjected to simultaneous access in a plurality of tasks. In order to prevent confusion caused by this simultaneous access, an indivisible operation for inhibiting execution of asynchronous processing caused by an interruption while a predetermined program is running, is used.

In order to realize the above indivisible operation, the following conventional methods are used.

1) An indivisible operation instruction required in a program to be executed is installed in each processor. A processor subjected to the indivisible operation must have a WCS (Writable Control Storage) or the like.

2) The processor occupies a shared memory bus prior to execution of the indivisible operation and frees the shared memory bus after the indivisible operation is completed. A program for executing this indivisible operation generally runs in a privileged mode.

3) A monitor loop using a semaphore operation

instruction or the like is formed by software, and exclusive control is performed. As compared with a time required to perform desired processing, a time for executing software processing for an indivisible operation is often prolonged dependent on an indivisible operation.

Strong demand has arisen for providing a parallel processing system for executing parallel processing including an indivisible operation which is free from limitations in an operation mode and a processor subjected to an indivisible operation and which can reduce the overhead.

It is an object of the present invention to provide a method and apparatus for controlling an indivisible operation in a parallel processing system.

According to one aspect of the present invention, there is provided a parallel processing system comprising: a plurality of processors; memory means for storing a plurality of instruction words, each having control information for an indivisible operation; and a bus for connecting the memory means to the plurality of processors, and wherein each of the processors includes: means for reading out the instruction word from the memory means though the bus, and obtaining the control information from the read out instruction word; first determination means for determining whether or not the indivisible operation is being performed in the

processor itself; second determination means for determining whether or not the indivisible operation is being performed in another processor without the processor; and means for controlling the indivisible operation in accordance with the obtained control information and results determined by the first and second determination means.

According to another aspect of the present invention, there is provided a method for controlling an indivisible operation in a parallel processing system having a plurality of processors, the method comprising the steps of: receiving an instruction word having at least one of control information for designating execution of instruction in an indivisible operation and control information for designating an end of the execution of the instruction in the indivisible operation; obtaining control information from the received instruction word: determining whether or not a desired processor executes the indivisible operation when the obtained control information is control information for designating the execution of the instruction in the indivisible operation; determining whether or not the indivisible operation is being performed in another processor without the desired processor when the desired processor does not execute the indivisible operation; and executing the indivisible operation in the desired processor when the indivisible operation is not performed in

another processor.

This invention can be more fully understood from the following detailed description when taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a block diagram showing a schematic arrangement of a parallel processing system according to an embodiment of the present invention;

Fig. 2 shows a format of the content of an instruction word used in parallel processing including an indivisible operation;

Fig. 3 is a block diagram showing the main part of a processor in the parallel processing system shown in Fig. 1; and

Fig. 4 is a flow chart for explaining operations of an indivisible operation control circuit in the processor in the parallel processing system shown in Fig. 1.

A preferred embodiment of the present invention will be described with reference to the accompanying drawings.

Fig. 1 is a block diagram showing an arrangement of a parallel processing system according to an embodiment of the present invention. Referring to Fig. 1, the parallel processing system comprises processors 1 and 2, a shared memory 3, a shared memory bus 6, and an indivisible operation control line 9.

The shared memory 3 stores an instruction word (to be described later) as a parallel processing program or

the like shared by the processors 1 and 2.

The processors 1 and 2 comprise shared memory bus control circuits 4 and 5 and indivisible operation control circuits 7 and 8, respectively. The shared memory bus control circuits 4 and 5 perform occupying/release control of the shared memory bus 6 to access the shared memory 3. The indivisible operation control circuits 7 and 8 control indivisible operations in the processors 1 and 2.

The shared memory bus 6 is used to connect the shared memory 3 to the shared memory bus control circuits 4 and 5.

The indivisible operation control line 9 is set at high level when at least one of the processors 1 and 2 performs an indivisible operation. Otherwise, i.e., when the indivisible operation is not executed, the indivisible operation control line 9 is set at low level.

Fig. 2 shows a format of the content of the instruction word used in parallel processing including an indivisible operation. Referring to Fig. 2, the instruction word is constituted by a first control bit field A, a second control bit field B, an instruction code field C, and an operand field D. A first control bit a for designating instruction execution in the indivisible operation is included in the first bit field A. A second control bit b for designating the end of the

indivisible operation upon execution of the instruction is included in the second control bit field B. An instruction code c representing an arithmetic operation to be executed is included in the instruction code field C. An operand d representing an object subjected to the arithmetic operation is included in the operand field D.

In this embodiment, information bits associated with an indivisible operation, i.e., the first and second control bits are included in each instruction word. The indivisible operation is performed in accordance with these control bits. Each control bit is set at "0" or "1". When the first control bit is set at logic "1", it indicates that an indivisible operation is performed. When the second control bit is set at logic "1", it indicates that the indivisible operation is completed.

Fig. 3 is a block diagram showing the main part of the processor 1 in the parallel processing system shown in Fig. 1. Referring to Fig. 3, the processor 1 comprises a shared memory bus control circuit 4, an indivisible operation control circuit 7, an instruction decoder 11, a sequencer 12, and a flip-flop (FF) 13.

The shared memory bus control circuit 4 outputs an instruction word read out from the share memory 3 to the instruction decoder 11.

The instruction decoder 11 selects the content of the first control bit field A and the content of the

second control bit field B, i.e., the first and second control bits a and b from the instruction word output from the shared memory bus control circuit 4 and outputs the first and second control bits a and b to the indivisible operation control circuit 7. The instruction decoder 11 also selects the content of the instruction code field C and the content of the operand field D, i.e., the instruction code c and the operand d from the instruction word output from the shared memory bus control circuit 4 and outputs the instruction code c and the operand d to the sequencer 12.

The flip-flop (FF) 13 indicates the state of the current indivisible operation in the processor 1. That is, when an instruction in the indivisible operation is executed by the processor 1, the flip-flop 13 is in a set state. Otherwise, the flip-flop 13 is in a reset state.

The indivisible operation control circuit 7 controls the indivisible operation of the processor 1 on the basis of the values of the first and second control bits a and b from the instruction decoder 11, the set/reset state of the flip-flop 13, and the level status (low level/high level) of the indivisible operation control line 9.

The processor 2 shown in Fig. 1 has the same arrangement as that of the processor 1 and is not illustrated in Fig. 3.

An operation of the parallel processing system having the processor with the arrangement described above will be described below.

When a predetermined instruction word is read out from the shared memory 3 and is input to the instruction decoder 11 in the processor 1, the instruction decoder 11 outputs the first and second control bits a and b included in the input instruction word to the indivisible operation control circuit 7. At the same time, the instruction decoder 11 outputs the instruction code c and the operand d included in this input instruction word to the sequencer 12. The sequencer 12 generates a control signal g on the basis of the instruction code c and the operand d in accordance with an operation stop signal e from the indivisible operation control circuit 7. A predetermined instruction is executed by this control signal g. Note that the operation stop signal e is used to stop the operation of the sequencer 12.

The indivisible operation control circuit 7 controls the indivisible operation of the processor 1 in accordance with a flow chart of Fig. 4 on the basis of the value of the first control bit a, the value of the second control bit b, the set/reset state of the flip-flop 13, and the level state of the indivisible operation control line 9.

It is determined in step S1 whether the first control bit a and the second control bit b are received

from the instruction decoder 11. If YES in step S1, i.e., if the first and second control bits a and b are received, it is determined in step S2 whether the first control bit a is set at logic "1". That is, it is determined in step S2 whether an indivisible operation is performed.

If it is determined in step S2 that the first control bit a is not logic "1", that is, if the first control bit a is logic "0", an instruction is executed while the corresponding indivisible operation is not performed. It is then determined in step S1 whether the next instruction word is received.

If it is determined in step S2 that, however, the first control bit a is logic "1", it is determined that the indivisible operation is performed. It is then determined in step S3 whether the second control bit b is set at logic "1". That is, it is determined in step S3 whether the indivisible operation is ended.

If it is determined in step S3 that the second control bit b is not logic "1", that is, if the second control bit b is set at logic "0", it is determined that the indivisible operation is not ended. It is then determined in step S4 whether the flip-flop 13 is reset.

If it is determined in step S4 that the flip-flop 13 is reset, it is determined that execution of the indivisible operation must be assured. It is determined in step S5 whether the indivisible operation control

line 9 is set at low level.

If it is determined in step S5 that the indivisible operation control line 9 is set at low level, it is determined that an indivisible operation is not performed in other processors, e.g., the processor 2. The indivisible operation control line 9 is set at high level (step S6), and the flip-flop 13 is set (step S7). Thus, instruction execution in an indivisible operation is enabled in the processor 1. On the other hand, instruction execution in indivisible operations is inhibited in other processors.

In step S8, an input of the operation stop signal e to the sequencer 12 is stopped. The control signal g is generated by the sequencer 12, and instruction execution in the indivisible operation is started.

On the other hand, if it is determined in step S4 that the flip-flop 13 is not reset, i.e., the flip-flop 13 is set, execution of the indivisible operation in the processor 1 is determined to be already assured. The operations in steps S6 and S7 are not performed, and the input of the operation stop signal e to the sequencer 12 is stopped (step S8). Therefore, as described above, the control signal g is generated by the sequencer 12, thereby starting instruction execution in the indivisible operation.

If it is determined in step S5 that the indivisible operation control line 9 is not set at low level, i.e.,

if the indivisible operation control line 9 is set at high level, it is determined that indivisible operation is performed in another processor. In this case, the operation stop signal e is input to the sequencer 12 (step S9). Therefore, the control signal g is not generated by the sequencer 12, and instruction execution in the indivisible operation is inhibited in the processor 1.

More specifically, since the value of the received first control bit a indicates that an indivisible operation is performed, the indivisible operation must be performed in the processor 1. However, since the indivisible operation control line 9 is set at high level, it is determined that instructions in the indivisible operation are executed in an other processor. Therefore, until instruction execution in the indivisible operation in the other processor is completed, instruction execution in the indivisible operation is inhibited in the processor 1.

It is determined in step S10 whether an external interruption signal h is input to the indivisible operation control circuit 7 while instruction execution in the indivisible operation is kept inhibited in the processor 1. If it is determined in step S10 that the external interruption signal h is input, the input of the operation stop signal e to the sequencer 12 is stopped (step S11). Therefore, the instruction

execution inhibit state of the processor 1 is released. In addition, an interruption enable signal f is input to the sequencer 12 (step S12). At this time, the instruction is executed by the control signal g generated on the basis of the instruction code c and the operand d in the sequencer 12, and then interruption processing is performed.

When this interruption processing is completed, it is determined again in step S5 whether the indivisible operation control line 9 is set at a low level. More specifically, the level state of the indivisible operation control line 9 is monitored. The processor 1 is set in a wait state of instruction execution in the indivisible operation until the indivisible operation control line 9 is set at the low level.

If it is determined in step S3 that the second control bit b is set at logic "1", it is determined that the indivisible operation is ended after instruction execution by the instruction word including this second control bit b is performed. After the instruction is executed, the indivisible operation control line 9 is set at the low level (step S13), and the flip-flop 13 is reset (step S14). It is defined by a software that the first control bit a is set to be at logic "1" when the second control bit b is set at logic "1".

It is determined in step S15 whether the external interruption signal h is input to the indivisible

operation control circuit 7. If YES in step S15, the interruption enable signal f is input to the sequencer 12 to perform interruption processing (step S16).

When the external interruption signal h is input while the instruction in the indivisible operation is being executed in the processor 1, the interruption enable signal f for performing interruption processing is not input to the sequencer 12 until the instruction of the instruction word having the second control bit b of logic "1" is executed, the indivisible operation control line 9 is set at a low level (step S13), and the flip-flop 13 is reset (step S14).

In this manner, control of the indivisible operation in the processor 1 is performed. Note that control of an indivisible operation in the processor 2 can be performed as in the processor 1.

As has been described above, according to the parallel processing system of this embodiment, the indivisible operation control circuit independent from the shared memory bus control circuit for controlling the shared memory bus is arranged in each processor. The information bits associated with an indivisible operation, i.e., the first and second control bits are included in each instruction word. As is apparent from the flow chart (Fig. 4), every time each instruction word is read out from the shared memory, the values of the first and second control bits are determined. The

indivisible operation is controlled on the basis of these values. When an instruction which does not require an indivisible operation is to be executed, the shared memory is not occupied, and an indivisible operation can be performed by another processor. Therefore, parallel processing including an indivisible operation, which is free from limitations in an operation mode and a processor subjected to an indivisible operation can be performed with a reduced overhead.

The above embodiment exemplifies control of indivisible operations in two processors. However, the present invention is not limited to operations in two processors. The number of processors can be three or more in control of an indivisible operation.

The number of shared resources is not limited to one. An indivisible operation can be performed in a plurality of resources. In this case, the number of indivisible operation control lines is set in accordance with the number of resources, and in response to this, a plurality of indivisible operation control circuits must be arranged in each processor.

Claims:

1. A parallel processing system comprising:

a plurality of processors;

memory means for storing a plurality of instruction words, each having control information for an indivisible operation; and

a bus for connecting the memory means to the plurality of processors, and

wherein each of the processors includes:

means for reading out the instruction word from the memory means though the bus, and obtaining the control information from the read out instruction word;

first determination means for determining whether or not the indivisible operation is being performed in the processor itself;

second determination means for determining whether or not the indivisible operation is being performed in an another processor without the processor; and

means for controlling the indivisible operation in accordance with the obtained control information and results determined by the first and second determination means.

2. The system according to claim 1, wherein the control information includes at least one control bit for designating execution of an instruction in an indivisible operation and a control bit for designating an end of the execution of the instruction in the

indivisible operation.

3. The system according to claim 1, wherein the first determination means has a flip-flop, and determines whether or not the indivisible operation is being performed in the processor itself in accordance with a set/reset state of the flip-flop.

4. The system according to claim 1, wherein the second determination means has an indivisible operation control line, and determines whether or not the indivisible operation is being performed in the another processor in accordance with a level state of the indivisible operation control line.

5. The system according to claim 1, wherein when the indivisible operation is not performed in the processor, it is determined that whether or not the indivisible operation is being performed in another processor without the processor.

6. The system according to claim 5, wherein when the indivisible operation is not performed in another processor, the processor executes the indivisible operation.

7. The system according to claim 1, wherein the processor further comprises means for freeing and reserving the memory means.

8. A parallel processing system comprising:

a plurality of processors;

memory means for storing a plurality of instruction

words, each having at least one of control information for designating execution of an instruction in an indivisible operation and control information for designating an end of the execution of an instruction in an indivisible operation;

a bus for connecting the memory means to the plurality of processors, and

wherein each of the processors includes:

means for reading out the instruction word from the memory means though the bus, and obtaining the control information from the read out instruction word;

means for determining whether or not the indivisible operation is being performed in the processor itself;

means for determining whether or not the indivisible operation is being performed in another processor without the processor; and

means for controlling the indivisible operation in accordance with the obtained control information and determination results.

9. The system according to claim 8, wherein when the indivisible operation is not performed in the processor, it is determined that whether or not the indivisible operation is being performed in another processor without the processor.

10. The system according to claim 9, wherein when the indivisible operation is not performed in

another operation, the processor executes the indivisible operation.

11. A method for controlling an indivisible operation in a parallel processing system having a plurality of processors, the method comprising the steps of:

receiving an instruction word having at least one of control information for designating execution of instruction in an indivisible operation and control information for designating an end of the execution of the instruction in the indivisible operation;

obtaining control information from the received instruction word:

determining whether or not a desired processor executes the indivisible operation when the obtained control information is control information for designating the execution of the instruction in the indivisible operation;

determining whether or not the indivisible operation is being performed in another processor without the desired processor when the desired processor does not execute the indivisible operation; and

executing the indivisible operation in the desired processor when the indivisible operation is not performed in another processor.

12. A method for controlling an indivisible operation in a parallel processing system, substantially as hereinbefore described with reference to the

accompanying drawings.

13. An apparatus for controlling an indivisible operation in a parallel processing system, substantially as hereinbefore described with reference to the accompanying drawings.

Published 1992 at The Patent Office, Concept House, Cardiff Road, Newport, Gwent NP9 1RH. Further copies may be obtained from